
Coping with lists in the ifcOWL ontology

Pieter Pauwels, piipauwel.pauwels@ugent.be

Ghent University, Ghent, Belgium,

Walter Terkaj, walter.terkaj@itia.cnr.it

Institute of Industrial Technologies and Automation (ITIA), Milano, Italy

Thomas Krijnen, t.f.krijnen@tue.nl

Technical University of Eindhoven (TU/e), Eindhoven, Netherlands

Jakob Beetz, j.beetz@tue.nl

Technical University of Eindhoven (TU/e), Eindhoven, Netherlands

Abstract

Over the past few years, several suggestions have been made of how to convert an EXPRESS schema into an OWL ontology. The conversion from EXPRESS to OWL is of particular use to architectural design and construction industry, because one of the key data models in architectural design and construction industry, namely the Industry Foundation Classes (IFC) is represented using the EXPRESS information modelling language. In each of these conversion options, the way in which lists are converted (e.g. lists of coordinates, lists of spaces in a floor) is key to the structure and eventual strength of the resulting ontology. In this article, we outline and discuss the main decisions that can be made in converting LIST concepts in EXPRESS to equivalent OWL expressions. This allows one to identify which conversion option is appropriate to support proper and efficient information reuse in the domain of architecture and construction.

Keywords: Semantic Web, Industry Foundation Classes, Resource Descriptions Framework

1 Introduction

The representation of aggregation data types (e.g. lists) is still not explicitly addressed in the Web Ontology Language (OWL - Hitzler et al. (2012)), even if it plays a key role in several applications. Other ontology languages, such as the Knowledge Interchange Format (KIF¹ - Genesereth and Fikes (1992)), or modelling languages like EXPRESS (ISO 10303-11, 2004) propose built-in solutions to deal with lists and sets of data. These data structures are particularly relevant also to support the conversion of data models formalised using other languages into an OWL ontology, so that the data model integration features offered by semantic web technologies (Berners-Lee et al., 2001) can be exploited. An example is represented by the conversion of an EXPRESS schema into an OWL ontology. This is of

particular use to architectural design and construction industry, because one of the key data models in architectural design and construction industry, namely the Industry Foundation Classes (IFC), is represented using the EXPRESS information modelling language. An OWL version of IFC, named ifcOWL, would provide a number of advanced semantic technologies to experts in architectural design and construction that are currently seldom available or used in this domain, e.g. reasoning capabilities (inference) and semantic search (Beetz et al., 2009; Schevers and Drogemuller, 2005; Agostinho et al., 2007; Pauwels et al., 2011; Pauwels and Van Deursen, 2012).

The IFC EXPRESS schema (Liebich et al., 2013) includes several aggregation types, which are intensively used to describe geometry, among others. In the case of geometry, the EXPRESS schema relies mainly on the LIST data type, which accommodates an ordered aggregation. Also property sets and relational

¹ <http://logic.stanford.edu/kif/specification.html>

objects often use aggregation data types (in those cases unordered SET data types) to link one type instance to a number of other type instances. There are a number of ways in which such aggregation types can be converted into an OWL expression. In the case of SET data types, it is typically most appropriate to convert the instances of these SET data types into a number of non-functional data property instances, as is proposed by Pauwels and Terkaj (2015a, 2015b). However, the conversion of ordered collections (LIST and ARRAY data types) often results in representations of an undesirable form, making the resulting RDF graphs following an ifcOWL specification hard to use. In particular, using the simplest conversion routine based on the RDF terms `rdf:list` - `rdf:first` - `rdf:rest` (Brickley and Guha, 2014; Dean and Schreiber, 2004) results in an ontology that is in the OWL Full profile (see Hitzler et al. (2012)). An ontology that is in the OWL Full profile cannot be used to exploit the desirable inference and semantic search functionalities in the domain of architectural design and construction. Alternative options, in which a custom `ifc:list` construct is used, typically result in similar complex and verbose graph structures. In this paper, we review the conversion options suggested in literature, and highlight the relevance of the problem for the architectural design and construction industry. Additionally, we introduce two new options to approach the problem, aiming to find an appropriate way to cope with this issue in the usage of IFC and ifcOWL.

2 The ifcOWL ontology

Building information modelling (BIM) is one of the most notable efforts in recent years regarding information management in construction industry (Eastman et al., 2008). BIM environments allow one to semantically describe any kind of information about the building in one information model. The Industry Foundation Classes (IFC) standard (ISO 16739, 2013), developed and maintained by the buildingSMART² organisation, aims at supporting these activities by providing a central “conceptual data schema and an exchange file format for BIM data” (Liebich et al., 2013). Using the IFC data model and instance serialisation formats, BIM data can be

exchanged between software applications covering a wide range of use cases (e.g. 4D planning, 5D cost calculation, CFD simulation, structural analysis).

The IFC data model is represented as a schema in the expressive EXPRESS data specification language defined in ISO 10303-11:2004 that “consists of language elements which allow an unambiguous data definition and specification of constraints on the data defined and by which aspects of product data can be specified” (ISO 10303-11, 2004). The EXPRESS data specification language is very powerful since it allows one to define in detail data types, entities, attributes, cardinality restrictions, type restrictions, complex rule expressions, complex derivation expressions, and complex functions. As a single element can be described in diverse ways by each BIM environment, the export and import possibilities to and from IFC should guarantee that each BIM environment is able to map its own descriptions to a generally understandable IFC format, thus improving interoperability of information.

The semantic web initiative (Berners-Lee et al., 2001) shares some of the formal specification goals that IFC is targeting for in construction industry. Namely, it allows one to describe shared conceptualisations of information in a flexible and generic language that takes the form of directed labelled graphs. Each node in such a graph represents a concept or object in the world and each arc in this graph represents the logical relation between two of these concepts or objects. A graph can be constructed using the Resource Description Framework (RDF) data model (Schreiber and Raimond, 2014). RDF graphs can be given an improved semantic structure using RDF vocabularies or ontologies (RDFS, Brickley and Guha (2014); and OWL, Hitzler et al. (2012)).

There is a considerable parallel between the buildingSMART effort towards the specification and standardisation of IFC for construction industry, and the W3C effort towards the specification and standardisation of RDF for web data. On a schema level, EXPRESS has several similarities to OWL and IFC instance files result in graph-like structures that make RDF a suitable capturing format. However, there are differences between the languages. For instance, Barbau et al. (2012) emphasise the lack of formal semantics in EXPRESS, arguing that a logic-based language, such as OWL, brings a number of modelling advantages in knowledge

² <http://www.buildingsmart.com/>

representation and semantic data sharing. Additionally, Beetz et al. (2009) stress the limits of EXPRESS with respect to the reuse of existing ontologies and interoperability with semantic web tools. On the other hand, EXPRESS provides complex elements which are not commonly available in regular OWL language profiles. For example, EXPRESS explicitly allows one to represent 'enumeration data types', 'select data types', a considerable number of well-specified 'aggregation data types' (namely, bags, sets, lists and arrays), and complex procedural statements (WHERE rules, RULE declarations and FUNCTION declarations). So,

Listing 1: EXPRESS specification of IfcCartesianPoint.

```

1 ENTITY IfcCartesianPoint
2   SUBTYPE OF (IfcPoint);
3   Coordinates : LIST [1:3] OF IfcLengthMeasure;
4   DERIVE
5     Dim : IfcDimensionCount := HIINDEX(Coordinates);
6   WHERE
7     CP2Dor3D : HIINDEX(Coordinates) >= 2;
8 END_ENTITY;
```

both languages have unique features, strengths, weaknesses and distinct levels of expressiveness.

The many parallels between the data modelling approaches in IFC and RDF make it feasible to publish IFC instance data as RDF graphs and thus harnessing the benefits and features of both languages. The formalisation of IFC data as RDF graphs requires the conversion of the IFC schema, defined as an EXPRESS schema, to an OWL ontology. Various research initiatives proposed conversion approaches from an EXPRESS schema to an OWL ontology. Schevers and Drogemuller (2005) proposed a first unidirectional conversion map from EXPRESS to OWL, taking IFC as a reference example and highlighting some of the key issues to be addressed. Agostinho et al. (2007) proposed a generic mapping between EXPRESS and OWL within a broader Model Morphism initiative. Beetz et al. (2009) proposed a semiautomatic method for converting EXPRESS schemas to OWL files in order to enhance the applicability and re-usability of the IFC standard. Barbau and colleagues specified a set of generic rules for the automated conversion from EXPRESS to OWL within the OntoSTEP initiative (Krima et al., 2009; Barbau et al., 2012).

Another conversion tool was presented by Pauwels and Van Deursen (2012).

Currently, two initiatives have been set up to further develop the specification of an ifcOWL ontology, for its use in the context of the architectural design and construction industry. The first initiative is the Linked Building Data community group hosted at the World Wide Web Consortium (W3C)³. This group aims at listing and specifying the use cases around building data in RDF, thus helping the development of the required ontologies, including ifcOWL. The second initiative is the Linked Building Data working group, which is part of the 'Technical Room' within BuildingSMART⁴. This group aims primarily at providing and maintaining a recommended ifcOWL ontology as an additional BuildingSMART specification.

3 The aggregation data types in the EXPRESS schema of IFC

The conversion of EXPRESS aggregation data types plays an important role in the generation of the ifcOWL ontology. In particular, the LIST and ARRAY data types in EXPRESS represent ordered collections of elements (ISO 10303-11, 2004; p. 23). In contrast to an ARRAY, which is always fixed in size, a LIST can be unbounded. Furthermore, it is possible to nest LIST data types, thus creating two- or more-dimensional aggregations (LIST OF LIST). The EXPRESS current schema of IFC4 is limited to one- and two-dimensional aggregations.

As a reference, we will consider the example of the IfcCartesianPoint entity in the IFC4 schema (see Listing 1). IfcCartesianPoint has an attribute named Coordinates (line 3 in Listing 1), which refers to a LIST of at least one and at most 3 instances of IfcLengthMeasure. The WHERE rule given in Listing 1 (line 7) additionally specifies that the highest index (HIINDEX) of the Coordinates property should be 2 or more. This implies that the Coordinates property should refer to a

³ <http://www.w3.org/community/lbd/>

⁴ <http://www.buildingsmart.org/lbd>

LIST of 2 or 3 instances of IfcLengthMeasure (2D versus 3D).

The importance of IfcCartesianPoint in an IFC model can be appreciated by considering an example, representing the Book Tower in Ghent (Belgium)⁵. This IFC model counts 417033 entity instances, of which 45085 are instances of IfcCartesianPoint (10,6%). There are numerous other lists in the IFC model, therefore the conversion of lists (and other aggregation

fragments of OWL ontology in the following sections, while referring to the namespaces defined in Listing 2. Note that the 5 namespaces in lines 5 to 9 in Listing 1 are dummy namespaces that are used to distinguish between the ontologies and approaches used in the following sections of this paper. Only one of them should eventually be used and it should reside under namespace `www.buildingsmart-tech.org/ifcOWL#`. Also the namespace at line

Listing 2: Definition of namespaces used in the remainder of this paper.

```
1 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
2 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
3 @prefix owl: <http://www.w3.org/2002/07/owl#> .
4 @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
5 @prefix ifcowl1: <http://www.buildingsmart-tech.org/ifcOWL1#> .
6 @prefix ifcowl2: <http://www.buildingsmart-tech.org/ifcOWL2#> .
7 @prefix ifcowl3: <http://www.buildingsmart-tech.org/ifcOWL3#> .
8 @prefix ifcowl4: <http://www.buildingsmart-tech.org/ifcOWL4#> .
9 @prefix ifcowl5: <http://www.buildingsmart-tech.org/ifcOWL5#> .
10 @prefix ifcinst:
    <http://www.linkedbuildingdata.net/20150504_ListInstances#> .
```

data types) has a major impact on the eventual graph, in particular in terms of semantic precision and computational performance. Lengthy representations of LIST types result in very precise and correct definitions, but they require notably more time to be loaded in-memory and used. If performance matters significantly, then it might be a better option to use less precise representations of the same information that can be loaded considerably faster.

4 Conversion Procedures

In the following sections, we investigate a number of conversion procedures. We hereby distinguish between:

- Conversion to the regular `rdf:List` concept natively available in RDF
- Conversion to general purpose list concepts
- Conversion to customised concepts
- Conversion to customised concepts referring to portions of Well-Known Text (WKT)

The existing conversion procedures listed in the previous section typically take the second approach. The Turtle syntax (Beckett and Berners-Lee, 2011) will be used to present the

10 is a dummy namespace that represents the instance graph of a BIM model in IFC.

4.1 Procedure 1: standard RDF lists

Lists (or any ordered sequence) cannot be easily represented in an RDF graph, because RDF relies on a triple structure that inherently allows one to link only two concepts, not collections of multiple concepts. Lists are thus typically represented in RDF by linking each concept to the next using `rdf:List`, `rdf:first` and `rdf:rest` declarations (Brickley and Guha,

Listing 2: Example IfcCartesianPoint instance described according to the IFC EXPRESS schema.

```
1 #37=IFCCARTESIANPOINT((0.,0.,-350.));
```

2014; Dean and Schreiber, 2004). When using this construct in an OWL ontology, however, the OWL profile becomes OWL Full, thus impeding the use of reasoning engines that rely on Description Logics (DL – Baader and Nutt, 2003). In addition, the conversion into an OWL Full ontology results in verbose representations for the instantiations of the LIST concepts in the ontology. An instance of IfcCartesianPoint, which is expressed in a regular IFC STEP Physical File Format (SPFF) as in Listing 2, results in the expression given in Listing 3.

⁵http://smartlab1.elis.ugent.be:8889/IFC-repo/20110616_bookTowerGhent.ifc

One can clearly see that the OWL expression in Listing 3 is far more verbose and complex (24 lines equivalent to 21 triples) compared to the SPFF representation in Listing 2. Considering the number of lists that are defined in the IFC EXPRESS schema and instantiated in IFC files, the impact on complexity and data volume is high.

important additional features provided by OWL that is not available in regular other data modelling approaches. We distinguish four different proposals in defining such list concepts:

- the OWLList by Drummond et al. (2006)
- the Ordered List Ontology (OLO) by Abdallah and Ferris (2010)

Listing 3: Representation of the example `IfcCartesianPoint` instance in Listing 2, following the first conversion procedure documented here.

```

1  ifcinst:IfcCartesianPoint_37
2    rdf:type ifcowll:IfcCartesianPoint ;
3    ifcowll:Coordinates_of_IfcCartesianPoint
4      ifcinst:IfcLengthMeasure_List_371 .
5  ifcinst:IfcLengthMeasure_List_371
6    rdf:type rdf:List ;
7    rdf:first ifcinst:IfcLengthMeasure_371inst ;
8    rdf:rest ifcinst:IfcLengthMeasure_List_372 .
9  ifcinst:IfcLengthMeasure_371inst
10   rdf:type ifcowll:IfcLengthMeasure ;
11   ifcowll:has_double "0.0"^^xsd:double .
12 ifcinst:IfcLengthMeasure_List_372
13   rdf:type rdf:List ;
14   rdf:first ifcinst:IfcLengthMeasure_372inst ;
15   rdf:rest ifcinst:IfcLengthMeasure_List_373 .
16 ifcinst:IfcLengthMeasure_372inst
17   rdf:type ifcowll:IfcLengthMeasure ;
18   ifcowll:has_double "0.0"^^xsd:double .
19 ifcinst:IfcLengthMeasure_List_373
20   rdf:type rdf:List ;
21   rdf:first ifcinst:IfcLengthMeasure_373inst;
22   rdf:rest rdf:nil .
23 ifcinst:IfcLengthMeasure_373inst
24   rdf:type ifcowll:IfcLengthMeasure ;
25   ifcowll:has_double "-350.0"^^xsd:double .

```

4.2 Procedure 2: custom list concepts

An alternative option consists in defining a new general purpose LIST concept for OWL, as considered throughout most of the conversion procedures that were outlined in Sect. 2. By doing so, it is possible to remain in the OWL DL profile, instead of OWL Full, which allows one to use reasoning or inference engines. This is a major improvement over the above *Procedure 1*, because logical inference is one of the most

- the approach followed in OntoSTEP by Krma et al. (2009) and Barbau et al. (2012)
- the approach proposed in the ifcOWL ontology by Pauwels and Terkaj (2015a, 2015b)

These approaches are quite similar in handling lists. As an example, Listing 4 shows how the `IfcCartesianPoint` given in Listing 2 can be converted using the approach proposed by Pauwels and Terkaj (2015a).

Although the mentioned approaches are viable alternatives that allow one to build an ifcOWL ontology in OWL DL, their results do not improve *Procedure 1* in terms of compactness and computational efficiency. Indeed, the instance representation given in Listing 4 is about as long as the representation in Listing 3. The differences between Listing 3 and 4 are at line numbers 6 (see ifcowl:hasListContent), 7 (see ifcowl:hasNext), 13, 14, and 20. These particularly added properties replace the standard rdf:first, and rdf:rest statements in Listing 3 which would make the ifcOWL ontology OWL Full. The OWLList (Drummond et al., 2006) and OntoSTEP (Krima et al., 2009; Barbau et al., 2012) approaches make the same suggestion, even if using different names for the properties. The OLO approach (Abdallah and

for the specific list. For example, in the case of the IfcCartesianPoint (Listing 2), it is possible to add a number of new properties to directly refer to the single elements in a list of two or three coordinates (IfcLengthMeasure concepts), thereby implicitly allowing to distinguish between 2D and 3D IfcCartesianPoint concepts. The results of the application of the *Procedure 3a* is reported in Listing 5, where the object property ifcowl2:Coordinates_of_IfcCartesianPoint (line 3 in Listing 4) is replaced by three separate object properties that point directly to the appropriate IfcLengthMeasure instances (line 3-5 in Listing 5).

Furthermore, one might also design a procedure in which the *object* property ifcowl2:Coordinates_of_IfcCartesianPoint (line 3 in Listing 4) is replaced by three separate

Listing 4: Representation of the example IfcCartesianPoint instance in Listing 2, following the conversion procedure as documented in Pauwels and Terkaj (2015a)

```

1 ifcinst:IfcCartesianPoint_37
2   rdf:type ifcowl2:IfcCartesianPoint ;
3   ifcowl2:Coordinates_of_IfcCartesianPoint ifcinst:IfcLengthMeasure_List_371 .
4 ifcinst:IfcLengthMeasure_List_371
5   rdf:type ifcowl2:List ;
6   ifcowl2:hasListContent inst:IfcLengthMeasure_371inst ;
7   ifcowl2:hasNext inst:IfcLengthMeasure_List_372 .
8 ifcinst:IfcLengthMeasure_371inst
9   rdf:type ifcowl2:IfcLengthMeasure ;
10  ifcowl2:has_double "0.0"^^xsd:double.
11 ifcinst:IfcLengthMeasure_List_372
12  rdf:type ifcowl2:List ;
13  ifcowl2:hasListContent ifcinst:IfcLengthMeasure_372inst ;
14  ifcowl2:hasNext ifcinst:IfcLengthMeasure_List_373 .
15 ifcinst:IfcLengthMeasure_372inst
16  rdf:type ifcowl2:IfcLengthMeasure ;
17  ifcowl2:has_double "0.0"^^xsd:double .
18 ifcinst:IfcLengthMeasure_List_373
19  rdf:type ifcowl2:List ;
20  ifcowl2:hasListContent ifcinst:IfcLengthMeasure_373inst;
21 ifcinst:IfcLengthMeasure_373inst
22  rdf:type ifcowl2:IfcLengthMeasure ;
23  ifcowl2:has_double "-350.0"^^xsd:double .
.
```

Ferris, 2010) additionally allows one to directly define the index of each slot in the list; however, it is also possible to explicitly state the precedence relationship between slots and in this case OLO becomes very similar to the previously mentioned approaches.

4.3 Procedure 3: conversion into more meaningful concepts

A third option that could be considered is to add *entirely new* concepts that are customised

datatype properties, as shown in Listing 6. The datatype properties point directly to the appropriate literal values of type ifcowl4:IfcLengthMeasure (line 3-5 in Listing 6), which is then defined as an equivalent class (owl:equivalentClass) of xsd:double. Such a formal representation is a major improvement over the representations presented in Listing 3 and 4 (and 5) in terms of length, complexity and readability. The resulting RDF graphs following this conversion approach can thus be loaded

and used far more efficiently. In addition, the resulting RDF graph is also more meaningful, as the relations between the `IfcCartesianPoint` concept and its coordinates are made explicit.

However, it must be noted that the ifcOWL3 ontology and the ifcOWL4 ontology used in

4.4 Procedure 4: well-known text (WKT)

The suitability and usefulness of describing extensive collections of, for example, Cartesian points using notations compatible with DL mechanisms can be considered limited in many use cases. Namely, using general purpose

Listing 5: Representation of the example `IfcCartesianPoint` instance in Listing 2, following the conversion Procedure 3a that explicitly adds new semantic concepts

```
1 ifcinst:IfcCartesianPoint_37
2   rdf:type ifcowl3:IfcCartesianPoint ;
3   ifcowl3:xCoordinate_of_IfcCartesianPoint ifcinst:IfcLengthMeasure_371inst ;
4   ifcowl3:yCoordinate_of_IfcCartesianPoint ifcinst:IfcLengthMeasure_372inst ;
5   ifcowl3:zCoordinate_of_IfcCartesianPoint ifcinst:IfcLengthMeasure_373inst .
6 ifcinst:IfcLengthMeasure_371inst
7   rdf:type ifcowl3:IfcLengthMeasure ;
8   ifcowl3:has_double "0.0"^^xsd:double .
9 ifcinst:IfcLengthMeasure_372inst
10  rdf:type ifcowl3:IfcLengthMeasure ;
11  ifcowl3:has_double "0.0"^^xsd:double .
12 ifcinst:IfcLengthMeasure_373inst
13  rdf:type ifcowl3:IfcLengthMeasure ;
14  ifcowl3:has_double "-350.0"^^xsd:double .
```

Listings 5 and 6 have concepts that are beyond the original EXPRESS schema. Indeed, the new properties cannot be mapped to definitions in the EXPRESS schema where there is only ‘a list with ordered coordinates’, thus hindering the use of general purpose SPFF to RDF (and vice versa) converters. Moreover, *Procedure 3b* makes direct use of datatypes, thus discarding the ‘objectification’ pattern that was proposed in Schevers and Drogemuller (2005) and used in Beetz et al. (2009), Barbau et al. (2012), and Pauwels and Terkaj (2015a). This objectification pattern is required to correctly convert SELECT

inference mechanisms to address complex computational geometry problems will limit the performance or will be impossible without dedicated (procedural, non-DL) functions as provided by geometry kernel libraries. When interacting with such structures, most applications will map and transform them into more efficient, intermediary structures. Even if dedicated support for ordered collections in reasoning and query engines might be expected in the future, the verbosity of standard approaches discussed in sections 4.1 and 4.2 will most certainly hamper their efficient use at

Listing 6: Representation of the example `IfcCartesianPoint` instance in Listing 2, following a conversion Procedure 3b that explicitly adds new semantic concepts

```
1 ifcinst:IfcCartesianPoint_37
2   rdf:type ifcowl:IfcCartesianPoint ;
3   ifcowl4:xCoordinate_of_IfcCartesianPoint "0.0"^^ifcowl4:IfcLengthMeasure ;
4   ifcowl4:yCoordinate_of_IfcCartesianPoint "0.0"^^ifcowl4:IfcLengthMeasure ;
5   ifcowl4:zCoordinate_of_IfcCartesianPoint "-350.0"^^ifcowl4:IfcLengthMeasure .
```

data types in EXPRESS to equivalent expressions in OWL. Hence, there is a risk that the proposal in Listing 6 leads to infeasible representations of SELECT data types in OWL. Namely, it might become possible that a SELECT data type in OWL can be instantiated and refer to a SELECT data type which is at that point both a datatype (Listing 6) and an OWL class wrapping a data type (Listing 5), which is infeasible in OWL DL (see also Pauwels and Terkaj, 2015a).

larger scales on data volume to be processed alone.

Therefore, a fourth procedure considered here relies on the usage of well-known text (WKT - ISO/IEC 13249-3 (2011)). This proposal has been made in the geospatial domain⁶, where large amounts of geometrical information must

⁶

<http://www.geoapi.org/3.0/javadoc/org/opengis/referencing/doc-files/WKT.html>

be represented both efficiently (short) and in a semantically meaningful manner. In this proposal, numeric values that are part of a particular concept or entity (e.g. a Cartesian point) are represented using (1) a keyword in upper case followed by the parameters of the object in brackets. This procedure applied to the case of the `IfcCartesianPoint` results in the formal representation given in Listing 7.

This formal representation clearly is an even larger improvement over the representations presented in Listing 3 and 4 (and 5 and 6) in terms of length, complexity and readability. The advantage compared to *Procedure 3* is that this approach can likely remain closer to the original EXPRESS schema: there are no additional concepts and the instance representation in Listing 2 is highly similar to the instance representation in Listing 7. A possible loss in terms of semantic precision could be at least partially softened by formally defining a customised *datatype* that is pointed

approach would reduce the number of nodes in the RDF graph even more, as multiple `IfcCartesianPoint` instances would be folded into a single WKT string object. This procedure can be followed up to higher order faceted boundary representations and the `IfcTriangulatedFaceSet` that is newly introduced in IFC4.

However, note that, with the existing set of WKT definitions, the representations that can be expressed are limited to polyhedral surfaces. For a more advanced string serialisation of curved face boundaries, one could look, for example, at the SVG standard, but it is likely that serialising a full-fledged BRep model into a string is beyond the scope of an ifcOWL serialisation. In addition, needless to say, for entities that do not pertain to the geometrical nature of a product, for example the `IfcMaterialLayerSet`, which features an ordered list of `IfcMaterialLayers`, no equivalent WKT concepts can be found, nor

Listing 7: Representation of the example `IfcCartesianPoint` instance in Listing 2, following a conversion *Procedure 4* that relies on the use of WKT

```
1  ifcinst:IfcCartesianPoint_37
2  rdf:type ifcowl5:IfcCartesianPoint ;
3  ifcowl5:Coordinates_of_IfcCartesianPoint "POINTZ(0.0 0.0 -
350.0)"^^ifcowl5:wktLiteral .
```

to `ifcowl5:Coordinates_of_IfcCartesianPoint`, instead of purely relying on informal agreements between the end users. In the case of Listing 7, for example, one could use `ifcowl5:wktIfcLengthMeasure`, instead of `ifcowl5:wktLiteral`. However, it remains to be seen if the correctness of these datatype values can be checked by semantic inference engines. Yet, such a trade-off between semantic precision and computational efficiency needs to be considered for domains with lots of numeric value aggregations, such as the geospatial domain and the geometric parts of the construction industry.

The mapping to WKT offers a plausible alternative to coordinate components, because the equivalents for these concepts exist in the WKT specification. Higher order geometrical representations also have WKT equivalents. For example, the `IfcPolyline` entity, a representation item consisting of a single attribute typed as a LIST of `IfcCartesianPoint`, can accurately be expressed as a LINESTRING WKT. Such an

would they be meaningful as flattening them into textual strings would hinder graph-based querying of these concepts.

5 Conclusion

As indicated at the outset of this paper, information technologies (IT) for the domain of architectural design and construction typically relies on a lot of aggregations (lists, sets, arrays, bags) for the representation of numeric values. A clear example is the intensive use of LIST expressions in IFC files. Indeed, the EXPRESS schema of IFC heavily relies on LIST data types. These data types do not map well on formal representations in OWL, simply because there is limited support for ordered aggregation types in OWL.

When considering *Procedure 3a* and *3b* as part of the same procedure, we presented four different procedures that can be used to represent aggregation types in OWL for the particular case of the Industry Foundation Classes (IFC). From the outline of these four procedures, it is clear that a trade-off needs to be made between semantic precision and

computational efficiency. The procedures that result in the semantically most precise representations (*Procedure 1* and *2*) also result in lengthy and overly complex representations, yet they are the closest equivalents to the original EXPRESS schema. The procedures that result in computationally more efficient representations (*Procedure 3a*, *3b*, and *4*), on the other hand, could result in sloppier data representations, which could in turn result in errors, flaws and mistakes in the applications relying on that data. Yet, as many have concluded in the

geospatial domain, this might be the only option to make the use of RDF graphs and OWL ontologies feasible for the architectural design and construction industry.

In further research, we will apply the two last conversion procedures to all the applicable data types in the IFC schema and analyse what this results in for an appropriate number of sample IFC files. This research will allow one to fully appreciate the viability of both approaches as alternatives to the conversion procedure that is currently used (‘2).

References

- Abdallah, S.A., and Ferris, B. (2010). The Ordered List Ontology 0.72 - Namespace Document 23 July 2010, <http://purl.org/ontology/olo/core#>, accessed April 2015.
- Agostinho, C., Dutra, M., Jardim-Goncalves, R., Ghodous, P., Steiger-Garcia, A. (2007). EXPRESS to OWL morphism: making possible to enrich ISO10303 modules. In: Loureiro, G., and Curran, R. (Eds.). Complex Systems Concurrent Engineering, London: Springer, pp. 391–402.
- Baader, F., and Nutt, W. (2003). Basic description logics. In: Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (Eds.). Description Logic Handbook: Theory, Implementation, and Applications, Cambridge: Cambridge University Press, pp. 47–100.
- Barbau, R., Krima, S., Rachuri, S., Narayanan, A., Fiorentini, X., Fofou, S., Sriram, R.D. (2012). OntoSTEP: Enriching product model data using ontologies, Comput. Aided Design 44, pp. 575–590.
- Beckett, D., Berners-Lee, T. (2011). Turtle - Terse RDF Triple Language - W3C Team Submission 28 march 2011, W3C Team Submission, <http://www.w3.org/TeamSubmission/turtle/>, accessed April 2015.
- Beetz, J., Van Leeuwen, J., and de Vries, B. (2009). IfcOWL: a case of transforming EXPRESS schemas into ontologies, Ai. Edam. 23, pp. 89–101.
- Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web, Sci. Am. 284, pp. 35–43.
- Brickley, D., and Guha, R.V. (2014). RDF Schema 1.1 - W3C Recommendation 25 February 2014, W3C Recommendation, <http://www.w3.org/TR/rdf-schema/>, accessed April 2015.
- Dean, M., and Schreiber, G. (2004). OWL Web Ontology Language Reference - W3C Recommendation 10 February 2004, W3C Recommendation, <http://www.w3.org/TR/owl-ref/>, accessed April 2015.
- Drummond, N., Rector, A., Stevens, R., Moulton, G., Horridge, M., Wang, H., Sedenberg, J. (2006). Putting OWL in order: Patterns for sequences in OWL. In: OWL Experiences and Directions (OWLED 2006).
- Eastman, C.M., Teicholz, P., Sacks, R., and Liston, K. (2008). BIM handbook: a guide to building information modeling for owners, managers, architects, engineers, contractors, and fabricators. Hoboken: Wiley.
- Genesereth, M.R., and Fikes, R.E. (1992). Knowledge Interchange Format Version 3.0 Reference Manual, Technical Report Logic-92-1, Logic Group at the Computer Science Department of Stanford University, <http://logic.stanford.edu/kif/specification.html>, accessed April 2015.
- Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P.F., and Rudolph S. (2012). OWL 2 Web Ontology Language Primer (Second Edition) - W3C Recommendation 11 december 2012, W3C Recommendation, <http://www.w3.org/TR/2012/REC-owl2-primer-20121211/>, accessed April 2015.
- International Organization for Standardization (2004). ISO 10303-11: Industrial automation systems and integration - Product data representation and exchange - Part 11: Description methods: The EXPRESS language reference manual, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=38047, accessed April 2015.
- International Organization for Standardization (2011). ISO/IEC 13249-3:2011 Information technology - Database languages - SQL multimedia and application packages - Part 3: Spatial, http://www.iso.org/iso/iso_ca-talogue/catalogue_tc/catalogue_detail.htm?csnumber=53698, accessed April 2015.
- International Organization for Standardization (2013). ISO 16739: Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries, http://www.iso.org/iso/catalogue_detail.htm?cs-number=51622, accessed April 2015.
- Krma, S., Barbau, R., Fiorentini, X., Sudarsan, R., Sriram, R. (2009) OntoSTEP: OWL-DL ontology for STEP, National Institute of Standards and Technology, Technical Report NISTIR 7561.

- Liebich, T., Adachi, Y., Forester, J., Hyvarinen, J., Richter, S., Chipman, T., Weise, M., Wix, J. (2013). Industry Foundation Classes IFC4 official release, <http://www.buildingsmart-tech.org/ifc/IFC4/final/html/index.htm>, accessed April 2015.
- Pauwels, P., and Van Deursen, D. (2012). IFC/RDF: adaptation, aggregation and enrichment. In: Report of the First International Workshop on Linked Data in Architecture and Construction, pp. 2–5.
- Pauwels, P., and Terkaj, W. (2015a). EXPRESS to OWL for construction industry: towards a recommendable and usable ifcOWL ontology, *Automat. Constr.* (submitted).
- Pauwels, P., and Terkaj, W. (2015b). ifcOWL: The EXPRESS to OWL conversion pattern, <https://www.w3.org/community/lbd/ifcOWL/>, accessed April 2015.
- Schevers, H., and Drogemuller, R. (2005). Converting the Industry Foundation Classes to the Web Ontology Language. In: Proceedings of the 1st International Conference on Semantics, Knowledge and Grid, pp. 556–560.
- Schreiber, G., and Raimond, Y. (2014). RDF 1.1 Primer - W3C Working Group Note 24 june 2014, W3C Working Group Note, <http://www.w3.org/TR/2014/NOTE-rdf11-primer-20140624/>, accessed April 2015.